EVERNODE

Any DApp. For Anyone. Forever.
A permissionless, flexible, scalable Layer 2 smart contract network composed from the XRP Ledger.

by

Scott Chamberlain and Richard Holland

27 July 2021

1. INTRODUCING EVERNODE

1.1 We propose Evernode, a project to bring "layer 2" smart contracts to the XRP Ledger. It combines our HotPocket protocol with the proposed Hooks amendment - small pieces of code that allow XRP Ledger Accounts to function as "lite smart contracts" - to create a global, permissionless, decentralised network of nodes designed and incentivised to run any dApp cheaply and speedily in any language at any scale in concert with the XRP Ledger.

The Promise of dApps

1.2 DApps - decentralised applications - are a new form of software that runs on blockchains without centralised ownership or control. DApps are more secure and reliable than traditional applications. This permissionless, censorship-resistant future is a key building block for a future that promises mainstream access to self-sovereign identity, improved privacy, decentralised finance, and tokenised art, rights, and assets.

Early DApp Platforms

1.3 Early dApp platforms have struggled to scale due to high fees, slow transaction times, and inflexibility as to language and functionality, all legacies of the blockchains on which they run. These kinds of drawbacks have hampered the mainstream adoption of dApps and the realisation of the benefits they offer.

1.4 These limitations are particularly acute if the dApp automates legal obligations or relationships. Lack of scaling inevitably means centralised "off-chain" computation or storage for what is otherwise a decentralised system.

Evernode

1.5 Evernode overcomes these drawbacks so everyone can benefit from the full variety of dApps. It has four main components:

(a) **Consensus Protocol:** HotPocket is a Unique Node List (UNL) based consensus protocol that allows multiple machines to become a mini-blockchain by enforcing consensus rules on inputs and outputs and maintaining a shared, canonical state.

(b) **Native Currency:** Evers, the native digital currency of Evernode, is an XRP Ledger token distributed by a Hook on the XRP Ledger to reward Nodes for participating in the network and used by dApps to pay hosting fees to Nodes.

(c) **Evernode Hook:** A Hook on the XRP Ledger that controls the emission of Evers and confirms each Node's membership in Evernode, their willingness to host dApps, and their agreement to be paid in Evers.

(d) **DEX**: The XRP Ledger's native decentralised exchange which Evernode uses to automate the buying and selling of hosting services with Evers.

1.6 In concert, these components create Evernode: a reliable, secure, and decentralised "network-of-networks" of HotPocket Nodes piggybacking off the XRP Ledger that anyone can join, capable of running a wide variety of cheap, speedy dApps that anyone can use, anytime.

Governance Game

1.7 An inherent feature of Evernode would be the ability for holders of Evers to compete via a Governance Game over the key parameters of the Evernode Hook, including the distribution of Evers, the identification of Nodes, and the rules of the Governance Game itself.

Ecosystem Curator

1.8 Evernode would have a formal Not-For-Profit Foundation to help curate its nascent ecosystem and act as a trusted repository of the open-source code.

2. HOTPOCKET - EVERNODE'S BEATING HEART

2.1 As previously detailed, the HotPocket Consensus Protocol is a consensus engine that facilitates convenient smart contract development. It uses a cooperative UNL-based consensus algorithm akin to the Ripple Consensus Protocol, abstracted for any transaction or input type.

2.2 It has five main components:

(a) **UNL Consensus Protocol:** HotPocket is a Unique Node List (UNL) based consensus protocol that allows multiple Linux machines to become a mini-blockchain by enforcing consensus rules on inputs and outputs and maintaining a shared, canonical state.

(b) **Rapid State Sync:** A bundle of additional features designed to make it as easy as possible to spin up a HotPocket Node and sync it to the existing network.

(c) **Contract Lifecycle Management:** HotPocket ensures all contracts have the same configuration and allows contracts to be upgraded automatically at consensus.

(d) **Daemon:** HotPocket clusters can be "nailed" to any layer 1 blockchain through a daemon, called "Sashimono", that co-ordinates the cluster through encrypted messages posted to the layer 1 blockchain.

(e) **Minimal setup:** Hot Pocket enables new nodes to join an existing contract with minimal known information to sync the new node with the cluster.

2.3 In concert, these components provide an "out of the box" solution for a network of HotPocket Nodes running a mini-blockchain capable of supporting a wide variety of cheap, speedy dApps coordinated or composed by any layer 1 chain.

3. EVERS - EVERNODE'S NATIVE CURRENCY

3.1 Evers are the proposed native currency of Evernode. They would be tokens that live on the XRP Ledger, and their distribution would be governed by a new form of mini-smart contract on the XRP Ledger called a Hook. Hooks are further detailed at https://hooks-testnet.XRP Ledger-labs.com/

3.2 Evers allow for automated buying and selling of hosting services from Evernode Nodes. Without a native digital currency, (and the related features of Hosting Tokens and the DEX), this process would be prohibitively cumbersome or hopelessly centralised.

Evers - The Details

Name: *Evers*

Ticker: *EVRS*

Smallest Unit: *a "Now". 1 Now = 0.00000001 Evers.*

Distribution Mechanism: *Airdrops plus Rewards to reliable nodes*

Reward Trigger: *Every Moment = every 72 ledgers (est. six minutes)*

Maximum Supply: *25,804,800*

Emission Timeframe: *~59.61 years*

Evers - The Rewards Emission Schedule

3.3 No pre-sale or ICO is planned. The protocol would be launched, fully-functioning without pooling of any funds for development. Evers would be either gifted (because they may be worth nothing, especially early on) or earned as rewards for running a reliable Node, a synthetic form of mining.

3.4 We propose to distribute Evers over 10 Epochs, with each Epoch involving the distribution of 2,580,480 Evers.

(a) **Epoch 1 – Development:** 2,580,480 Evers to developers upon launch of TestNet.

(b) **Epoch 2 – Testing:** 2,580,480 Evers distributed to those who help develop and test Evernode by running nodes and launching dApps.

(c) **Epoch 3 – Launch:** On launch, 2,580,480 Evers will be airdropped to XRP holders like the Flare Networks airdrop (ie, in proportion to all holders through wallets and participating exchanges while excluding accounts Ripple and its founders); and

(d) **Epochs 4-10 – Hosting Rewards:** In the final phase, the Hook that controls Evers will create and distribute new Evers as rewards shared equally between all reliable Nodes. Rewards will start at 64 Evers per Moment and halve every Epoch. The first Epoch is roughly 24 weeks and each subsequent Epoch doubles in duration. Rewards cease after ten Epochs. Distribution is skewed to favour early-adopters, since each new node is relatively more valuable to the network when it is young and small.

Proposed Evernode Reward Schedule

The Rules: *In the first Epoch, rewards are shared between developers. In the second Epoch, rewards are shared between beta testers. In the third Epoch, rewards are airdropped to most XRP holders. After the first three Epochs, Rewards are distributed every Moment. A Moment is 72 ledgers (est. 6 minutes). Rewards are shared equally among eligible nodes. Rewards start at 64 Evers per Moment and halve every Epoch. The first Epoch lasts 12 weeks. Each subsequent Epoch doubles in duration. Rewards cease at the end of the tenth Epoch.*

Epoch	**Epoch Duration (est. Weeks)**	**Reward Trigger (Mins./Moment)**	**Hosting Rewards**					**Elapsed Years**
			Per Moment	**Per Hour**	**Per Day**	**Per Week**	**Total For Epoch**	
1st	Airdrop	Developers	N/A	N/A	N/A	N/A	2,580,480	N/A
2nd	Airdrop	Testers	N/A	N/A	N/A	N/A	2,580,480	N/A
3rd	Airdrop	XRP Holders	N/A	N/A	N/A	N/A	2,580,480	N/A
4th	24	6	64	640	15,360	107,520	2,580,480	0.46
5th	48	6	32	320	7,680	53,760	2,580,480	0.92
6th	96	6	16	160	3,840	26,880	2,580,480	1.85
7th	192	6	8	80	1,920	13,440	2,580,480	3.69
8th	384	6	4	40	960	6,720	2,580,480	7.38
9th	768	6	2	20	480	3,360	2,580,480	14.77
10th	1536	6	1	10	240	1,680	2,580,480	29.54
TOTALS	**3048**						**25,804,800**	**58.61**

Table 1- Evernode Reward Schedule

Evers – Use Case

3.5 Evers would be instantly useful as a means of exchange for hosting services on Evernode. They would be valuable to the extent that Evernode's dApp hosting services are valuable. Evers may or may not have any value outside the network, but Evernode functions whether Evers have an external monetary value or not.

3.6 People should run a Node because they want to see Evernode thrive, and they should want to own Evers to fuel their dApps to deploy and run on Evernode. Many useful and valuable (and profitable) dApps and businesses can be built on Evernode independent of any external value Evers might have.

4. THE EVERNODE HOOK - EVERNODE'S HOSTING DIRECTORY

4.1 The Evernode Hook also manages the registry of Nodes that have signalled their desire to be on Evernode.

4.2 The Evernode Hook would be configured to broadcast the Node's details, the fact it wishes to be on Evernode, its availability to host dApps, and its willingness to be paid in Evers. The Hook would track the Node's performance so dApps can see how reliable it is. This information would be accessed from any XRP Ledger-compatible wallet or browser plug-in.

Why Use the XRP Ledger and Hooks

4.3 The way HotPocket plus its daemon, Sashimono, is designed, Evernode could have been constructed as a stand-alone chain or "nailed" to any layer 1 chain with sufficient smart contract capabilities to issue tokens and function as a message board.

4.4 However, we think Evernode could be best launched as a layer 2 solution composed via the XRP Ledger because of the suite of benefits the XRP Ledger offers. These benefits come in two categories: benefits that come from being within the ecosystem of an existing chain; and benefits the XRP Ledger has over other layer 1 chains.

Existing Ecosystem Benefits

(a) **Integration:** By issuing Evers as a token on a layer 1 chain we avoid the need to build an independent ecosystem of wallets, browser plug-ins, and explorers. Evernode will interoperate with existing XRP Ledger tools, like Xumm. Exchanges that support XRP can easily support Evers.

(b) **No dUNL:** By issuing Evers on a layer 1 chain, we avoid the need for a separate Evernode network with a separate dUNL and all the complications that come with specifying, identifying, and incentivising a decentralised dUNL.

XRP Ledger Benefits vs Other Layer 1 Chains

(c) **Speedy:** The XRP Ledger is fast for a blockchain. Transactions are confirmed within 3-5 seconds, and it can handle up to 1500 transactions per second.

(d) **Cheap:** The XRP Ledger is cheap. Transactions cost a fraction of a penny.

(e) **Reliable:** The XRP Ledger is reliable. It has been running continuously for over 8 years and processed over 62 million ledgers without halting.

(f) **Secure:** The XRP Ledger is not vulnerable to double spending. If validators cannot reach consensus the ledger halts until human actors reconfigure the dUNL to remove the compromised nodes.

(g) **Native DEX:** The XRP Ledger has a native decentralised exchange. This is very useful for issuing and trading hosting tokens for Evers, a core feature of Evernode.

4.5 This suite of benefits, we believe, makes the XRP Ledger the ideal choice for Evernode's layer 1 chain with one caveat: the implementation of Hooks.

Necessity for Hooks

4.6 Hooks are described as follows:

> *"Hooks add smart contract functionality to the XRP Ledger: 'layer one' custom code to influence the behaviour and flow of transactions. Hooks are small, efficient pieces of code being defined on an XRP Ledger account, allowing logic to be executed before and/or after XRP Ledger transactions."*[1]

4.7 As small pieces of code that allow XRP Ledger Accounts to function as "lite smart contracts", Hooks allows for Evers to be minted and issued to reliable Nodes in accordance with the proposed emissions schedule, subject to changes that might be implemented through the Governance Game.

4.8 Without Hooks, the XRP Ledger lacks the capacity to support Evernode in a fully decentralised form. If the Hooks amendments does not proceed, we would have to pivot to an alternative chain, such as Ethereum or, more likely (because of Ethereum's cost and slowness) Binance Smart Chain.

5. THE DEX - EVERNODE'S AUTOMATED HOSTING MARKET

5.1 Evernode would leverage the XRP Ledger's native decentralised exchange (DEX) to allow Nodes and dApps to exchange Evers and Hosting Tokens.

Hosting Tokens

5.2 Nodes would mint Hosting Tokens on the XRP Ledger. A Hosting token is a "best-endeavours" promise to provide a minimum level of computation resources (CPU and Memory per Moment) to the holder, plus a price list of additional computation resources the dApp can purchase from the Host with Evers.

SAMPLE - PROPOSED EVERNODE HOSTING RULES

Background: *We exchange hosting tokens for Evers. Evers may have no extrinsic worth and the point of Evernode is to host dApps on multiple Nodes so that one Node's failure doesn't matter so much.*

No Guaranteed Redemption: *We may refuse to redeem a hosting token for any reason or for no reason, without refund.*

Sole Remedy: *If we do not redeem a hosting token, your sole remedy is to resell the token on the DEX and/or to find an alternative Evernode host.*

Best Endeavours Only: *We do not promise services of any particular standard or availability, if available at all, and any services we do provide are strictly on a "best endeavours" basis.*

No Liability: *To the extent permitted by law, we offer no warranty, and deny any liability, in respect of our hosting services (except for fraud).*

Privacy: *We will not use, store, or reveal any personal information, except as necessary to provide hosting services to dApps on Evernode (a public blockchain network) or as required by law.*

[1] https://XRP Ledger-hooks.readme.io

5.3 These Tokens would be listed on the DEX where dApps can exchange them for Evers. DApps then redeem their Hosting Tokens with the Node for the specified computational resources (and pay for additional resources in Evers) as and when required or trade those Tokens back on the DEX.

5.4 Thus, the DEX functions as a way of taking a non-fungible service - hosting services on a particular server - and making it fungible by allowing everyone on Evernode to trade in and out of Hosting Tokens and Evers until they have the assets they desire.

The DEX and XRP

5.5 To trade and redeem Hosting Tokens with a given host, a dApp would have to open a trustline with that Host on the XRP Ledger. This requires a stake of 5 XRP, which is returned if the trustline is closed (i.e., the dApp no longer wishes to use that host).

The DEX's Quality Control Function

5.6 One important benefit of the DEX is its inherent capacity to police the quality of the Nodes.

5.7 The best signal of a Node’s trustworthiness is the natural price discovery of its Tokens on the DEX. Bad Nodes should quickly be exposed, and their Tokens shunned or under-priced relative to peers. Thus, the DEX provides an important circuit-breaker function, allowing poor quality Nodes to be easily auto-identified by dApps and dApp developers.

6. BENEFITS - GREATER FLEXIBILITY, MORE USEFUL DAPPS, ON DEMAND

6.1 HotPocket dApps don’t **run on** blockchains, they **are** blockchains. Each dApp is its own blockchain with its own chain history and dedicated nodes, making them incredibly flexible.

6.2 DApps may be public or private. DApps may call external services, read and write data directly to disk and the web, and generally perform any task a regular program can, without centralisation or trusted third parties and without requiring the programmer to implement their own consensus mechanisms.

6.3 This flexibility solves many problems that limit mass adoption of dApps including:

(a) **Privacy Compliance:** dApps can encrypt data, run only on hosts in a given jurisdiction, or only on hosts that have agreed to meet privacy regulations.

(b) **Scale & Flexibility:** dApps can run on as few or as many hosts as the dApp developer desires from a cost and security perspective.

(c) **On-Demand Oracles:** dApps can elect a sub-set or jury of their own nodes to get data from off-chain, agree on the truth, and report to the rest of the chain as a bespoke, on-demand oracle.

(d) **Enhanced Security:** dApps can detect when a host has become compromised or untrustworthy, shut down that instance of the dApp, and reload it on another, more trusted node selected from the Registry.

6.4 The above are all benefits of HotPocket. The additional benefit of Evernode is that all these benefits become available “on demand” through a permissionless, global network of hosts without the dApp developer having to spin up and maintain their own cluster of HotPocket Nodes.

7. GOVERNANCE GAME – SELF-SOVEREIGN HOOK MAINTENANCE

7.1 Evernode uses a Hook on the XRP Ledger. We envisage allowing holders of Evers to compete over the configuration of the Hook through a Governance Game. This Governance Game would allow holders of Evers to propose and vote on the Hook’s parameters as an inherent feature of holding Evers.

What Holders Govern Via Governance Game

7.2 Through the Governance Game, holders of Evers propose and vote on several things related to the functioning of the Evernode Hook as follows:

(a) **Evers**: Regulate the distribution of Evers by:

(i) Adding or removing an address from a blacklist.

(ii) Freezing or unfreezing any number of Evers in an address.

(iii) Changing the Rewards per Moment.

(iv) Changing the blocks per Moment.

(v) “Hardening” the proof of hosting by nominating a new, further test for Nodes to be deemed trustworthy and receive Evers as rewards.

(vi) Nominating a new Hook.

(b) **Hosts:** Regulate the Registry of Hosts by:

(i) Adding or removing a Node from the Blacklist.

(ii) Adding or removing a Domain Notary authorised to confirm the authenticity of a Node’s claimed domain name.

(iii) Changing the canonical definition of a Hosting Token.

(iv) Changing the default text of the rules by which each Host agrees to offer hosting on Evernode.

(c) **Votes:** Regulate the Governance Game itself by:

(i) Proposing a new time limit for proposals to expire.

(ii) Changing the number of tokens required to cast a vote.

(iii) Changing the number of tokens required to make a proposal.

Governance Game Rules

7.3 The Governance Game would be played according to the following rules:

(a) Anyone holding 200,000 Evers may propose a Resolution provided they "lockup" their Evers for the purposes of voting.

(b) Anyone holding 20,000 Evers may vote on a Resolution provided they also "lockup" their Evers for the purposes of voting.

(c) Each 20,000 of Evers a user "locks up" entitles them to 1 vote.

(d) A Resolution is carried if 14 days (measured in blocks on the XRP Ledger) after the Resolution is proposed the number of votes cast in favour of the Resolution is double the number of votes cast against it.

(e) Any user can activate a carried Resolution if they pay the fees of implementing the Resolution.

(f) Any Resolution (even if carried) is automatically null and void if 7 days (measured in blocks on the XRP Ledger) after the Resolution was carried nobody has paid the fees to implement the Resolution.

(g) Anyone can cancel a lapsed Resolution.

Self-Sovereign Governance

7.4 The Governance Game thus allows holders of Evers – should they wish to do so - to compete over the settings of the Hook, ensuring its on-going viability and fitness for purpose.

8. THE FOUNDATION – EVERNODE'S CURATOR

8.1 We propose a not-for-profit, notionally called The Evernode Foundation Ltd, to help curate Evernode's nascent ecosystem.

The Foundation

8.2 The Foundation would be an Australian public company Limited by Guarantee, meaning it has members, not shareholders. It would be Not-for-profit. Its Constitution would prevent it from distributing dividends or surplus assets on winding-up to members. If the Foundation winds-up, its assets must be gifted to another organisation with same or similar objects.

Foundation Members

8.3 Other than Founding Members, membership would be open to such other classes of members as the Board determines. It is likely to develop membership packages for Node owners, dApp developers, and associate members like lawyers, accountants, and VCs, but it will take time for the rights and eligibility for those memberships to be finalised.

Foundation Board

8.4 The initial Board of the Foundation would comprise of the founding members interested in actively supporting the Foundation to achieve its objectives and who funded the Foundation's establishment. Subsequent Boards would be elected by the membership in accordance with the Constitution.

Foundation as Curator

8.5 The Foundation’s main purpose as regards Evernode would be to curate the nascent Evernode ecosystem. As curator, it would have four main functions:

(a) **Curate Copy of Evernode Code:** It would curate and distribute a trusted copy of Evernode’s open-source code in conjunction with community volunteers.

(b) **Run a Node:** The Foundation, with the help of volunteer members, would run a Node on Evernode.

(c) **Notary Service:** The Foundation would be the first (potentially of many) able to confirm on-chain the real-world identity of Hosts that want their identity confirmed to strengthen their trustworthiness.

(d) **Stimulate Enhancements:** The Foundation will have Evers because of its participation in the beta testing and running a Node. It would use these Evers principally as bounties to stimulate community development of feature enhancements to the Evernode code base.

9. PROPOSED FUNCTIONALITY AT LAUNCH

9.1 Evernode would be fully functional at launch. Anyone would be able to run a Node and start earning Evers. Anyone could deploy and run a dApp with their Evers.

10. NEXT STEPS

10.1 Before Evernode can exist:

(a) HotPocket must be finalised and work.

(b) Sashimono must be finalised and work.

(c) Hooks must be finalised and implemented.

(d) All three must work in concert.

10.2 The best way to help make Evernode real is to participate in testing and implementing Hooks on the XRP Ledger.